Population parametrization of costly black box models using iterations between SAEM algorithm and kriging[1]

Emmanuel Grenier[a,b], Celine Helbert[c], Violaine Louvet[c,b], Adeline Samson[d], Paul Vigneaux[a,b,*]

[a] Univ Lyon, ENS de Lyon, CNRS UMR 5669, UMPA. F-69364 Lyon, France
[b] INRIA, Numed Team
[c] Univ Lyon, CNRS UMR 5208, Ecole Centrale de Lyon, ICJ. F-69622 Villeurbanne, France
[d] Universite Grenoble Alpes, LJK, F-38000 Grenoble, France. CNRS, LJK, F-38000 Grenoble, France

Emmanuel.Grenier@ens-lyon.fr, Celine.Helbert@ec-lyon.fr, louvet@math.univ-lyon1.fr, adeline.leclercq-samson@imag.fr, Paul.Vigneaux@math.cnrs.fr

Abstract

In this article we focus on parametrization of black box models from repeated measurements among several individuals (population parametrization). We introduce a variant of the SAEM algorithm, called KSAEM algorithm, which couples the standard SAEM algorithm with the dynamic construction of an approximate meta model. The costly evaluation of the genuine black box is replaced by a kriging step, using a basis of precomputed values, basis which is enlarged during SAEM algorithm to improve the accuracy of the meta model in regions of interest.

Keywords: Parameters estimation; SAEM algorithm; Kriging; Non-linear Mixed Effect Models; Partial Differential Equations; KPP equation

MSC: 65C60 65C40 62M05 60G15 65M32 65N21 35K57

[1] Computational and Applied Mathematics [ISSN: 0101-8205 (Print)]. Accepted March 25, 2016. DOI: 10.1007/s40314-016-0337-5 This is authors' version on HAL.

1 Introduction

In this article, we are concerned with the parametrization of models of the form

$$y = f(t, Z) + \varepsilon$$

where y is the observable, t is the time of observation, Z the individual parameters and ε is a measurement error term. The model f is referred to as a "black box" model. It may be a system of ordinary differential equations, of partial differential equations, or a multi agents system, or any combination of these model types. We will assume that it is costly, namely that its evaluation is very long. For instance one single evaluation of a reaction-diffusion equation in a complex geometry may last a few minutes or even a few hours if the coefficients are large or small, leading to a stiff behavior.

In this paper we focus on population parametrization from observations of f along time among N individuals. From these repeated longitudinal data, we search the distribution of the parameters Z in that given population of individuals. To take into account the various sources of variabilities (inter-individual and intra-individual variabilities), we use a non-linear mixed effect model.

The non-linear mixed effect model links the j-th measure, $j = 1, \ldots, N_i$, y_{ij} at times t_{ij} for individual $i = 1, \ldots, N$ with the black box model:

$$y_{ij} = f(t_{ij}, Z_i) + \varepsilon_{ij}, \tag{1}$$

where Z_i are p-vectors of the random individual parameters, ε_{ij} are random measurement errors, independent of the individual parameters Z_i. The errors ε_{ij} are assumed to be Gaussian

$$\varepsilon_{ij} \sim_{iid} \mathcal{N}(0, \sigma_\varepsilon^2). \tag{2}$$

The main difficulty is to identify the individual parameters Z_i, which are unknown and should be inferred from the observations. Identifying (or estimating) directly the Z_i from the data (y_{ij}) might be difficult when N_i is small, typically smaller or of the same order than p, the length of Z_i. To avoid this problem of dimension, we assume that the parameters Z_i are random and follow a given probability law determined by what we call later on population parameters. These population parameters are unknown and

the objective is to identify them rather than all the Z_i. In the following we assume that the parameters follow a Gaussian distribution

$$Z_i \sim_{iid} \mathcal{N}(\mu, \Omega), \tag{3}$$

where μ is a p-vector of expectation and Ω is a $p \times p$ matrix of covariance. The population parameters are $(\mu, \Omega, \sigma_\varepsilon^2)$.

We then look for parameter mean μ and variances Ω, σ_ε^2 which maximize the likelihood of the observations (y_{ij}). Once we have an estimation of μ and Ω from the observations (y_{ij}), we may want to estimate or approximate the individual variables Z_i which are the more likely given the observations for individual i and the distribution of individual parameters in the population $\mathcal{N}(\mu, \Omega)$.

SAEM algorithm is a classical approach to evaluate and estimate numerically the population parameters μ, Ω and σ_ε^2 from a non-linear mixed effect model (Kuhn and Lavielle, 2005). This algorithm requires a large number of evaluations of the model f, typically a few hundreds of thousands, or a few millions. If the model is costly, the total time of SAEM algorithm may be very huge, of a few days or even months.

A natural way to make SAEM doable with costly f is to replace it by an "approximate" model which in turn is much faster to compute. Such approximation is called *meta model* in the following, and we assume that under an appropriate asymptotic procedure it converges to the original model f. To build such a meta model, there exist numerous methods: discretizing the parameters space and using classical interpolation, reduced basis methods, etc. Schilders et al. (2008); Haasdonk and Lohmann (2011); Patera and Rozza (2006)
For instance, the first attempt of using such meta model with SAEM to decrease its computation time was presented in Grenier et al. (2014): parameter space (Z) is discretized with an inhomogeneous grid adapted to the variations of f and the meta model is given by a linear interpolation made on this fixed grid. This general method was illustrated on a reaction-diffusion partial differential equation showing that the SAEM computation time can be lowered from 23 days to 25 minutes. However this method is still subject to the classical "curse of dimensionality": one can reasonably only operate with a maximum of 5 or 6 parameters for the black box model.

One way to improve this problem is to use a more parsimonious interpolation such as kriging. Indeed, the kriging approach (where f is thought as

the realization of a Gaussian process (Sacks et al., 1989; Santner et al., 2003; Fang et al., 2005)) is less sensitive to dimension. Interestingly, kriging to build a fixed grid used by SAEM was later studied in Barbillon et al. (2015). They proved the convergence of the SAEM algorithm to the maximum likelihood of an approximate non-linear mixed effect model. It is also shown theoretically that the error produced by the kriging approximation can be controlled depending on the quality of the kriging grid. Therefore in practice, for a costly black box model f, we have to choose a kriging approximation with sufficient accuracy (depending on the available computational power). However it is more delicate to refine the mesh where the model really changes since it is not possible to rapidly identify where f has sharp transitions.

But we need to keep in mind that we deal here with a coupling between SAEM and the meta model, i.e. that actually, this meta model only needs to be precise in the regions of the parameters space which will be explored by the SAEM iterations. Based on this, Grenier et al. (2014) already proposed the methodology of a meta model which is refined during the SAEM algorithm itself, meaning that more points are added in the "grid" (or basis) of the meta model dynamically.
The aim of the present paper is thus to describe precisely and implement this idea of interactive coupling between the SAEM and the meta model building based on the kriging approach. The expected gain of this new algorithm, called KSAEM for "Kriging SAEM", is the following:

- since the meta model is dynamic, the off line step (i.e. building the meta model before SAEM) does not need to be very precise: as a consequence the initial meta model is obtained with only a few calls to the resolution of f;
- then during the SAEM (on line step), the meta model will be refined, only if one detects that the precision is not sufficient (in a sense defined later): as a consequence few other costly resolutions of f will be done, but most of the time only fast interpolations on the existing basis will be used;
- overall, the total number, say n_c, of costly evaluations of f for this SAEM run is lower than a precise off line building of a meta model.

A remark must be given here: comparison of the global computation cost with a fixed grid approach like in Grenier et al. (2014); Barbillon et al. (2015), can

not be done directly on n_c since a fixed grid is done once for all SAEM runs, whereas a dynamic meta model is built at each use of a SAEM algorithm. For instance, if a meta model is used many times on various data sets, it could be better to use a fixed grid approach than a dynamic grid approach.

The paper is presented as follows. In Section 2, we recall the problem of maximizing the likelihood of a non-linear mixed model and the classical SAEM algorithm. In Section 3, we present our algorithm. We start by quickly describing the kriging and then introduce the new algorithm called KSAEM. The two last sections are devoted to two examples: theophylline degradation, and KPP model.

2 Maximization of the likelihood and exact SAEM algorithm

This section is devoted to a brief presentation of the likelihood in the case of non-linear mixed effects models and the standard SAEM algorithm that allows to compute the maximum of the likelihood, providing an evaluation of the population parameters.

2.1 Non linear mixed effects model

Let us start with the ideal case when enough data are available for the i^{th} individual. Then the individual parameters Z_i can be estimated maximizing the Gaussian density of the observations $(y_{ij})_j$ given the (hidden) individual parameter Z_i (Gaussian error (2)):

$$p\Big((y_{ij})_j|Z_i;\sigma_\varepsilon^2\Big) = \frac{1}{\sigma_\varepsilon^{N_i}\sqrt{2\pi}^{N_i}}\exp\Big(-\frac{1}{2\sigma_\varepsilon^2}\sum_{j=1}^{N_i}(y_{ij}-f(t_{ij},Z_i))^2\Big).$$

This is equivalent to the classical least squares minimization procedure (non-linear regression)

$$\hat{Z}_i = \operatorname{argmin}\sum_{j=1}^{N_i}(y_{ij}-f(t_{ij},Z_i))^2.$$

However in many interesting cases, only few data are collected per individual, and the nonlinear regression procedure is useless. An alternative is to pool all

the data together, and to calibrate and estimate the distribution of individual parameters in the whole population, assuming they have a Gaussian distribution through the non-linear mixed effect model. Individual parameters are recovered in a second part.

Let us denote $\theta = (\mu, \Omega, \sigma_\varepsilon^2)$ the population parameters. The density of the individual parameters Z_i is simply

$$p\Big(Z_i;\theta\Big) = \frac{1}{\sqrt{(2\pi)^p \det(\Omega)}} \exp\Big(-\frac{1}{2}(Z_i-\mu)^t\Omega^{-1}(Z_i-\mu)\Big)$$

Hence for individual i, the joint density of observations $(y_{ij})_j$ and individual parameters Z_i is

$$p\Big((y_{ij})_j, Z_i;\theta\Big) = \frac{1}{\sqrt{(2\pi)^p \det(\Omega)}} e^{-\frac{1}{2}(Z_i-\mu)^t\Omega^{-1}(Z_i-\mu)} \frac{1}{\sigma_\varepsilon^{N_i}\sqrt{2\pi}^{N_i}} e^{-\frac{1}{2\sigma_\varepsilon^2}\sum_{j=1}^{N_i}(y_{ij}-f(t_{ij},Z_i))^2}$$

Assuming the N individuals to be independent, the density of the complete population variables $(y_{ij}, Z_i)_{ij}$ is

$$p\Big((y_{ij})_{ij}, (Z_i)_i;\theta\Big) = \Pi_{i=1}^N p\Big((y_{ij})_j, Z_i;\theta\Big)$$

As $(Z_i)_i$ are hidden variables, the density of the observations $(y_{ij})_{ij}$ given the parameters θ is the integral of $p\Big((y_{ij})_{ij}, (Z_i)_i;\theta\Big)$ with respect to Z_i:

$$g\Big((y_{ij})_{ij};\theta\Big) = \int p\Big((y_{ij})_{ij}, (Z_i)_i;\theta\Big) dZ_1 ... dZ_N. \tag{4}$$

With this expression we can define the log likelihood of θ to be

$$l(\theta) = \log g\Big((y_{ij})_{ij};\theta\Big).$$

The main problem is now to maximize this likelihood and to compute

$$\theta_\star = \text{argmax}_\theta\, l(\theta). \tag{5}$$

This problem is very delicate since the evaluation of a single value of l requires the evaluation of a multidimensional integral, which in turn requires numerous evaluations of our costly black box model. As stated, this is out of reach even for simple models. Several methods and algorithms have been proposed to solve this optimization problem. We focus in this paper on a stochastic version of the well-known EM algorithm (Dempster et al., 1977), namely the SAEM algorithm (Kuhn and Lavielle, 2005).

2.2 SAEM algorithm

The EM algorithm relies on a series of acute ideas and on two main iterative steps: the expectation step which computes a conditional expectation and the maximization step which maximizes the conditional expectation with respect to the parameters.

At iteration k of the EM algorithm, given the current value of the parameter θ_k, we proceed in two steps:

1. an expectation step computes, by "doubling" the parameter θ, the quantity

$$\begin{aligned} Q(\theta|\theta_k) &= \int \log p\Big((y_{ij})_{ij},(Z_i)_i;\theta\Big)p\Big((Z_i)_i|(y_{ij})_j;\theta_k\Big)dZ_1...dZ_N \\ &= \mathbb{E}\Big(\log p\Big((y_{ij})_{ij},(Z_i)_i;\theta\Big)|(y_{ij})_{ij};\theta_k\Big), \end{aligned} \tag{6}$$

   where $p\Big((Z_i)_i|(y_{ij})_{ij};\theta_k\Big)$ is the conditional density of the hidden variables Z_i given the observations $(y_{ij})_{ij}$;

$$p\Big((Z_i)_i|(y_{ij})_{ij};\theta_k\Big) = \frac{p\Big((y_{ij})_{ij},Z_i;\theta_k\Big)}{g((y_{ij})_{ij};\theta_k)} \tag{7}$$

   and g is the renormalization factor (the likelihood) defined by (4).

2. a maximization step updates the current value of the parameter

$$\theta_{k+1} = \text{argmax}_\theta\, Q(\theta|\theta_k). \tag{8}$$

It turns out that this maximization problem is much easier to compute since the integral in (6) is taken with respect to a fixed density (dependent of θ_k) which can be approximated through Monte Carlo procedure. Moreover, thanks to the log in the integrand, there is no exponential in function Q, which is simply bilinear in θ. The maximization procedure is therefore completely explicit provided we know how to compute the integral with respect to $dZ_1...dZ_N$.

To approximate this integral the EM algorithm is coupled to a Monte Carlo Markov Chain method (Kuhn and Lavielle, 2005). A Metropolis-Hastings algorithm provides a sequence of Markov chains $(Z_i^{kl})_l$ with stationary distribution $p((Z_i)_i|(y_{ij})_{ij};\theta_k)$. This is easily done using (7). Then

$Q(\theta|\theta_k)$ could be approximated by a Monte Carlo empirical mean using a large sample $((Z_i^{krl})_i)_{r=1,\dots,R_l}$ for L independent chains $l = 1, \dots L$. This leads to the Monte Carlo EM (Wei and Tanner, 1990).

$$\begin{aligned}\tilde{Q}(\theta|\theta_k) &= \frac{1}{L}\sum_l \frac{1}{R_l}\sum_r \log p\Big((y_{ij})_{ij}, (Z_i^{krl})_i; \theta\Big) = -\frac{N}{2}\log((2\pi)^p \det(\Omega)) \\ &\quad -\frac{1}{2}\sum_{i=1}^N N_i \log(2\pi\sigma_\varepsilon) - \frac{1}{L}\sum_l \frac{1}{R_l}\sum_i \frac{1}{2}(Z_i^{kl}-\mu)^t\Omega^{-1}(Z_i^{krl}-\mu) \\ &\quad -\frac{1}{L}\sum_l \frac{1}{R_l}\sum_i\sum_j \frac{(y_{ij}-f(t_{ij}, Z_i^{krl}))^2}{2\sigma_\varepsilon^2}.\end{aligned}$$

Note that, as stated previously, the maximization step (8) is completely explicit.

Note that the Monte Carlo approximation $\tilde{Q}(\theta|\theta_k)$ is somehow not adequate since we need to simulate a large number of Z_i^{kl} to get an accurate evaluation of $\tilde{Q}$. However as the EM algorithm goes on, θ_k converges, hence $\tilde{Q}(\theta|\theta_k)$ will be close to $\tilde{Q}(\theta|\theta_{k-1})$. To take advantage of this convergence, the idea of SAEM (Delyon et al., 1999) is to introduce Q_k defined iteratively, using only one realization Z_i^{kl} per chain by

$$Q_k = (1-\gamma_k)Q_{k-1} + \frac{1}{L}\sum_l \gamma_k \log p\Big((y_{ij})_{ij}, (Z_i^{kl})_i; \theta\Big)$$

where γ_k goes slowly to 0, and where Z_i^{kl} is one realization of Z_i under the conditional distribution $p((Z_i)_i|(y_{ij})_{ij}; \theta_k)$ for chain $l = 1, \dots, L$. Condition on γ_k are the following: $\sum_k \gamma_k^2 < \infty$ and $\sum_k \gamma_k = \infty$. A usual choice is thus $\gamma_k = \frac{1}{k^c}$ with $c \in]1/2, 1[$.

Delyon et al. (1999) and Kuhn and Lavielle (2005) prove the convergence of the sequence θ_k towards the maximum of the likelihood g, under smoothness assumptions on the likelihood function.

3 Coupling kriging and the SAEM algorithm

As presented before, in the SAEM algorithm, we need to evaluate f only in the Metropolis procedure to find out new sampling points Z_i^{kl}. The same evaluations of the model are then used in the computation of Q_k. This is a

costly step since it can not be parallelized for a given individual Z_i (though all individuals are independent and can be treated in parallel).

A natural idea is thus to replace f by a meta model f^{app}, that is by an approximation of f which is easy and fast to compute. One of the most popular meta model is Kriging (Sacks et al. (1989) and Santner et al. (2003)). It is largely used because of its flexibility and because at each point of the domain, it gives a variance of prediction that depends on the distance between the point and the observations. This approach is detailed in Section 3.1. Then we explain how we couple kriging with the SAEM algorithm in Section 3.2.

3.1 Kriging in a few words

Let us recall basic kriging for the function f. We work here at a fix time t. For simplicity of notations, we will not write the dependence in t in this Section. Assume that in a preliminary step, we have evaluated the function f exactly at several points z_j $(1 \leq j \leq n)$. Let us denote $D = \{z_1, \ldots, z_n\}$ this set of points, also called *kriging basis*. Note that the z_j are different from the observations (y_{ij}) and also from the (unknown) random individual parameters Z_j. We want to use the exact values of f at D to approximate f at another point z.

The idea is to suppose that the function f is the realization of a Gaussian process $(\Phi(z))_{z \in S}$, where $S \subset \mathbb{R}^p$, entirely defined by a mean function $m(.)$ and a covariance function $C(.,.)$. In the simplest case, m is assumed to be constant and the covariance function is assumed to be stationary with the form:

$$\forall z, z' \in S, C(\Phi(z), \Phi(z')) = \sigma^2 \prod_{\ell=1}^{p} k(|z_\ell - z'_\ell|; \beta_\ell)$$

where the parameter σ^2 corresponds to the overall variance on the domain S, β_ℓ to the correlation parameter in the ℓth direction and $k(.;.)$ is the correlation function. Different choices can be made for the correlation function depending on the expected regularity of f. For example, if f is highly regular, the Gaussian kernel $k(z, z'; \beta) = \exp(-\frac{|z-z'|^2}{\beta^2})$ is considered.

Given that probabilistic context, the kriging predictor f^{app} and the kriging variance Var^{app} are the expectation and variance of the process $\Phi(z)$ conditional to the exact values of f at points $D = \{z_1, \ldots, z_n\}$, i.e.

$$f^{app}(z) \quad = \quad m + c(z)C^{-1}(f_{1:n} - m\mathbf{1}_n) \tag{9}$$

$$Var^{app}(z) \quad = \quad \sigma^2 - c(z)C^{-1}c(z)^t \tag{10}$$

where $f_{1:n} = (f(z_j))_{j=1:n}$, $\mathbf{1}_n$ is a vector of ones, $C = (C(z_{j'}, z_j))_{j=1:n;j'=1:n}$ and $c(z) = (C(z, z_j))_{j=1:n}$.
The function $f^{app}(\cdot)$ is then the best approximation of f in the sense that it minimizes the mean quadratic error. The variance may be used as a quality indicator of the approximation of f by f^{app}. Looking carefully at the preceding formulas, it can be observed that the kriging predictor f^{app} is a linear combination of the exact values $(f(z_j))_{j=1:n}$. The weight of each exact value $f(z_j)$ in the prediction at point z strongly depends on $C(z, z_j)$, that is to say on the distance between the two points. The more z_j is close to z, the more influential is the corresponding observation in the prediction. Moreover, the predictor is strictly interpolating the observations, the variance is null at each observation point and increases with the distance to observation points.

In the following of the manuscript, the parameters m, σ^2 are considered known (respectively equal to 0 and 1) and all the correlation parameters $(\beta_\ell)_{\ell=1:p}$ are considered equal (case of geometric isotropy). Of note, the parameters could be estimated from the observations. Mean m and variance σ^2 are obtained by maximizing the likelihood function. The correlation parameters $(\beta_\ell)_{\ell=1:p}$ are also obtained by maximizing the likelihood function or by minimizing a cross validation criterion. Further, when trend parameters m are estimated, an additional variance is added to Var^{app} that takes into account the additional source of uncertainty coming from estimation procedure. The choice of the correlation function can also be discussed. Here the following Matern kernel has been used for its intermediate regularity (Santner et al. (2003)):

$$k(h;\beta) = \left(1 + \sqrt{5}\frac{h}{\beta} + \frac{5}{3}\left(\frac{h}{\beta}\right)^2\right) e^{-\sqrt{5}\frac{h}{\beta}}.$$

3.2 Iterations between SAEM and kriging

Now that the kriging approximation of f has been recalled, we present the idea of coupling SAEM and the kriging, to obtain a new algorithm called KSAEM.

3.2.1 Iterative kriging

In some cases the individual parameters Z_i will be concentrated on small areas of parameter space. In these cases we need to have a precise meta model in these areas, and do not need precise approximations of f away from these areas of interest. Of course if the individual parameters fill the whole parameter space, this observation is useless and what we propose will not improve very much the computation cost.

The main idea is to iteratively improve the meta model during the iterations of the SAEM algorithm. Each time we need to evaluate our model f at some new candidate point $\tilde{Z}$, we approximate this value by our meta model. Kriging gives an estimation $f^{app,k}(\tilde{Z})$ (9) based on the current kriging basis $D^k = \{z_1, \dots, z_{n_k}\}$ that contains n_k points and their corresponding exact evaluations $(f(z_j))_{j=1:n_k}$. We also obtain an estimate on the kriging error $Var^k(\tilde{Z})$ (10).

If the kriging error is small enough, we use $f^{app}(\tilde{Z})$ as a good approximation of f. If the kriging error is too large, we directly compute $f(\tilde{Z})$. This is a long step, but it increases the precision of our evaluation of f not only at $\tilde{Z}$, but also in the neighborhood of $\tilde{Z}$. As we assume that individual parameters Z_i are localized, we hope that this refinement will be used in future steps of SAEM, for larger k. We thus expect that this costly improvement will be used in the forthcoming steps of the algorithm.

As θ_k converges, we have a more and more precise idea of the areas of interest, and we can gradually improve our meta model in these areas, in order to decrease the approximation error $f - f^{app}$ in these areas. To decrease this error everywhere is useless since few individual parameters will be outside the areas of interest. To improve the approximation is costly, but hopefully will be focused on small areas, and of limited extent.

3.2.2 KSAEM algorithm

Let us now describe our algorithm called KSAEM. We choose a precision δ_k which slowly goes to 0. For notation's simplicity, we present the algorithm with one chain $L = 1$.

At iteration k of the SAEM algorithm, given the current value of the parameter θ_k, of the individual parameters $Z_i^{(k-1)}$ and the current kriging basis D^k, we proceed as follows:

- Simulation step: For each individual i, $i = 1, \dots, N$, (this step can be parallelized), we construct a sequence $Z_i^{k(m)}$ for $1 \leq m \leq M$, for some fixed M, starting from $Z_i^{(k-1)}$ and targeting the distribution $p(Z_i|(y_{ij})_j; \theta_k)$ (7), using a Metropolis-Hastings algorithm:
  - We simulate some new parameter $\tilde{Z}$ with a proposal law $q(\tilde{Z}, Z_i^{k(m-1)})$. We will not detail the proposals q here since they are exactly the same as in the classical SAEM algorithm.
  - We approximate $f(t_{ij}, \tilde{Z})$: two cases appear
    - Either $Var^k(\tilde{Z}) < \delta_k$. In this case we approximate $f(t_{ij}, \tilde{Z})$ by $f^{app,k}(t_{ij}, \tilde{Z})$.
    - Or $Var^k(\tilde{Z}) \geq \delta_k$. In this case we do compute $f(t_{ij}, \tilde{Z})$ exactly. We add $\tilde{Z}$ and $f(t_{ij}, \tilde{Z})$ to our kriging basis: $D^{k+1} = D^k \cup \{\tilde{Z}\}$ and include them for any further computation. This updates $f^{app,k}$, by progressive inclusion of new points.
  - Using this evaluation of $f(t_{ij}, \tilde{Z})$, we compute the acceptance probability:

$$\alpha(\tilde{Z}, Z_i^{k(m-1)}) = \min\left\{1, \frac{p(\tilde{Z}, (y_{ij})_j|\theta_k)}{p(Z_i^{k(m-1)}, (y_{ij})_j|\theta_k)} \frac{q(Z_i^{k(m-1)}, \tilde{Z})}{q(\tilde{Z}, Z_i^{k(m-1)})}\right\}, \tag{11}$$

  - We define $Z_j^{k(m)} = \tilde{Z}$ with probability $\alpha(\tilde{Z}, Z_i^{k(m-1)})$ and else $Z_j^{k(m)} = Z_i^{k(m-1)}$. After M iterations, we set $Z_j^k = Z_i^{k(M)}$.
- Stochastic Approximation step: We update Q_k (6)

$$Q_k = (1 - \gamma_k)Q_{k-1} + \gamma_k \log p\Big((y_{ij})_{ij}, (Z_i^k)_i; \theta\Big)$$

- Maximization step: Computation of θ_{k+1} as for the usual SAEM algorithm.

In the following applications we choose a piecewise decreasing profile for δ_k but this choice can be discussed. And its influence can be of importance for the sequential process.

The complete study of the convergence of KSAEM is beyond the scope of this paper.

4 A first example: theophylline degradation

4.1 Model

To illustrate our method, let us begin with a very simple case: the pharmacokinetic (PK) degradation of theophylline, an anti-asthmatic drug. This example is widely discussed in Monolix Team (2012).

The general context is the following: subject i receives an initial dose B at time 0 and serum concentrations (y_{ij}) are measured at times (t_{ij}). The degradation is modeled by a first-order (absorption) one-compartment model

$$y_{ij} = \frac{B k_{a_i} k_{e_i}}{C_{l_i}(k_{a_i} - k_{e_i})}\left(e^{-k_{e_i} t_{ij}} - e^{-k_{a_i} t_{ij}}\right) + \varepsilon_{ij}, \tag{12}$$

where C_{l_i} is the clearance, k_{a_i} and k_{e_i} are the absorption and elimination rates of subject i. The vector of individual parameters is

$$\phi_i = (\log(k_{a_i}), \log(k_{e_i}), \log(C_{l_i})).$$

We assume ϕ_i to be a log-normal random variable with a diagonal variance matrix, namely

$$\begin{aligned} \log(k_{a_i}) &\sim \mathcal{N}(\mu_{k_a}, \omega^2_{k_a}) \\ \log(k_{e_i}) &\sim \mathcal{N}(\mu_{k_e}, \omega^2_{k_e}) \\ \log(C_{l_i}) &\sim \mathcal{N}(\mu_{C_l}, \omega^2_{C_l}) \end{aligned}$$

Note that the model depends on four parameters: k_a, k_e, C_l and on time t, and is simply

$$f(k_a, k_e, C_l, t) = \frac{B k_a\, k_e}{C_l(k_a - k_e)}\left(e^{-k_e t} - e^{-k_a\, t}\right).$$

Three possibilities appear to incorporate time in our analysis. The first and the simplest is to compute kriging on three parameters k_a, k_e, C_l for each time step (see Marrel et al. (2011) for the spatial case). Another one is to consider time as an additional parameter and to use kriging algorithm on the four parameters k_a, k_e, C_l and t. The difficult aspect of this approach is to choose a good covariance function (Picheny and Ginsbourger (2013)). The last option is to consider that there are only three parameters k_a, k_e, C_l and that the output of the model is the function $t \to f(k_a, k_e, C_l, t)$. The

usual idea is to reduce the functional output space to a vectorial one by decomposing the functional output on a functional basis. Each coefficient from decomposition is approached by a kriging metamodel (see Auder et al. (2012) and Moutoussamy et al. (2015)). Here, the difficulties are the choice of an appropriate functional basis (wavelets, splines etc.), the choice of the right number of functions in the basis and the modelling of the possible dependance between coefficients. In our examples, we always consider the first approach, a little bit more time consuming than the others, but that gives good results.

4.2 Data analysis

We simulate 100 sets of data with $N = 100$ subjects, 12 points per subjects following the design of the original set of data (see Monolix Team (2012) for more details). The parameter values are set as follows: $\mu_{k_a} = 1.5, \mu_{k_e} = 0.08, \mu_{C_l} = 0.04$, $\omega^2_{k_a} = 0.01$, $\omega^2_{k_e} = 0.01$, $\omega^2_{C_l} = 0.01$ and $\sigma^2_\varepsilon = 0.55$. We want to compare the behavior of KSAEM and SAEM algorithms when estimating these 7 parameters. Algorithms are initialized with $\mu_{0,k_a} = 1, \mu_{0,k_e} = 0.025, \mu_{0,C_l} = 0.05$, $\omega^2_{0,k_a} = 1$, $\omega^2_{0,k_e} = 1$, $\omega^2_{0,C_l} = 1$ and $\sigma^2_{0,\varepsilon} = 1$.

For the SAEM algorithm we use standard parameters (300 iterations during the first phase of the algorithm, 300 in the second phase, $L = 3$ chains and two iterations per kernel). During its run, SAEM evaluates about 850000 times the model. Note that the model is so simple that its precise evaluation is very quick. As a consequence, SAEM algorithm converges within a few minutes on current laptops.

For KSAEM, we use a Matern kernel, with $\beta = 0.5$. We do not change the kernel during the run. This could be done in order to update the correlation function while new model values are evaluated (the study of this improvement is postponed to later works). We first evaluate the model on 30 points uniformly spread out over the parameters domain. The choice of δ_k is somehow arbitrary. If δ_k decreases too fast, the error of approximation of the meta model becomes much smaller than the error of SAEM algorithm. This does not improve the quality of the result of KSAEM, but leads to a large number of evaluations of the complete model, and thus to large computational times. If δ_k decreases too slowly, the error of approximation of the meta model is too large, and the results of KSAEM are of poor quality. For this example, after a few trials we have chosen to improve one hundred times the quality of kriging approximation with respect to the quality of the

Parameters	True values	SAEM	KSAEM
μ_{k_a}	1.5	1.476 (0.265)	1.837 (0.733)
μ_{k_e}	0.08	0.081 (0.015)	0.088 (0.027)
μ_{C_l}	0.04	0.040 (0.004)	0.041 (0.011)
$\omega^2_{k_a}$	0.01	0.040 (0.084)	0.023 (0.031)
$\omega^2_{k_e}$	0.01	0.041 (0.090)	0.011 (0.010)
$\omega^2_{C_l}$	0.01	0.010 (0.004)	0.013 (0.011)
σ^2_ε	0.55	0.040 (0.084)	0.024 (0.032)

Table 1: Simulation study with PK model (Theophylline): results obtained from 100 repetitions, with $N = 100$ individuals with the exact SAEM and KSAEM (SAEM coupled with a kriging approximation of the model). Results are presented in means and standard deviation in brackets.

initial kriging grid. We therefore choose δ_k as follows. During the first 50000 evaluations of the model, we add a new point to the kriging if the variance is larger than 0.05. Between 50000 and 500000, we add a new point if the variance is larger than 5.10^{-3} and after 500000 if the variance is larger than 5.10^{-4}.

At the end of the run, KSAEM has evaluated about 850000 times the approximate meta model through kriging, and has added 63 new kriging points. In this case KSAEM needs 93 evaluations of the complete model. This has to be compared with the 850000 evaluations of the model by SAEM. Hence, in this particular case, KSAEM needs close to 10000 times less evaluations of the complete model than SAEM needs.

Estimation results are presented in Table 1. The mean parameters μ_{k_a}, μ_{k_e} and μ_{C_l} are very well estimated with SAEM. The variance parameters are more difficult to estimate but this is well-known in this model. The estimations of the mean parameters by KSAEM are not as good as for SAEM (which is in agreement with the fact that a meta model is used instead of the complete model) but they are reasonably accurate.

5 A second example: KPP

5.1 Model

As a second illustration of KSAEM we will consider the following classical version of the KPP (or reaction-diffusion) equation

$$\partial_t u - \nabla.(\nu \nabla u) = \lambda u(1-u) \tag{13}$$

where $u(x)$ is the unknown concentration (assumed to be initially a compact support function for instance), ν the diffusion coefficient and λ the reaction rate. These equations are posed in a domain Δ with Neumann boundary conditions. Initially we assume that the support of u is very small and located at some point $x_0 \in \Delta$. We will assume that we do not observe u directly but only the total amount of u, namely

$$S(t) = \int_\Delta u(t,x)\ dx.$$

For instance, this equation can be used to model tumor growth. Often we do not have the precise location of the tumor (or do not want to enter into the difficulties of imaging) and only have an estimate of tumor size.

We will use this model in one space dimension. This model already retains some difficulties of partial differential equations based models (importance of space dimension, slow numerical evaluation) but not the difficulties of geometry which appear in two or three space dimensions. Note that even with refined numerical methods, if the diffusion ν is small and the reaction λ is large, equation (13) is stiff and therefore long to evaluate numerically. The evaluation time may be hundred or thousand times longer than for a classical ordinary differential equations system.

If we try to apply directly SAEM algorithm on this model, we have to deal with a very large computational cost (3 days on a laptop). In a former paper Grenier et al. (2014), we coupled SAEM algorithm with a precomputation step on a grid. This already speed up the algorithm, however the precomputation step can be shortened by our approach (see below). This model is therefore a good benchmark for population parametrization of complex systems.

5.2 KSAEM

We assume that the three parameters of our model x_0, λ and ν follow a log-normal distribution. The values used for the simulations are the following: $\mu_\lambda = 0.0236, \mu_\nu = 8.195 \times 10^{-7}$ and $\mu_{x_0} = 0.4$. The variances are all chosen equal $\omega_\lambda^2 = \omega_\nu^2 = \omega_{x_0}^2 = 0.04$ and the measurement noise is set to $\sigma_\varepsilon^2 = 0.05$. We generate randomly 100 sets of data with 100 individuals and simulate the complete model for these individuals. We then run both the SAEM and the KSAEM algorithms on these observations to compare the estimation of the 7 population parameters. Algorithms are initialized with $\mu_{0,\lambda} = 0.008, \mu_{0,\nu} = 3 \times 10^{-7}$, $\mu_{0,x_0} = 0.8$, $\omega_{0,\lambda}^2 = \omega_{0,\nu}^2 = \omega_{0,x_0}^2 = 1$ and $\sigma_{0,\varepsilon}^2 = 1$.

The SAEM algorithm requires about 900000 evaluations of KPP, namely almost one million evaluations of a partial differential equation which may be stiff if the diffusion is small. This lasts about 3 days on a laptop with our current implementation (C++ code).

For KSAEM, we use the Matern kernel with $\beta = 0.99$. We start with 20 evaluations of KPP. We will use a uniform 20 points basis for the kriging before running SAEM. It is also possible to start from optimized sets of points (Dupuy et al. (2015), Pronzato and Muller (2012) and Jin et al. (2005)), however this leads to bad results, in coherence with the theoretical results established in Barbillon et al. (2015) for other models. The sequence δ_k is chosen in order to increase the accuracy of the meta model by a factor 100. To this end, we update the kriging mesh before 50000 iterations if the variance is larger than 0.05, between 50000 and 500000 if the variance is larger than 5.10^{-3} and after 500000 if the variance is larger than 5.10^{-4}. After this whole process, only 22 points have been added in the basis, while the KSAEM run made about 930000 calls to the approximate meta model through kriging. Therefore the total number of KPP evaluations needed by KSAEM is 42. This is to be compared with the 900000 evaluations of KPP needed by SAEM. In this case KSAEM is much faster than SAEM since the kriging evaluation time is much smaller than the KPP evaluation time. A typical KSAEM run is about 1 minute (vs. 3 days for SAEM).

We compare the results obtained with KSAEM and with the exact SAEM. Results are presented in Table 2. The exact SAEM gives very good results for all the parameters, except σ_ε^2 which is slightly underestimated. KSAEM gives good results to estimate the three mean parameters λ, ν and x_0. The three individual variances ω_λ^2, ω_ν^2 and $\omega_{x_0}^2$ are overestimated, especially the first two. Thus, iterative kriging seems to be a good strategy, both saving

Parameters	True values	SAEM	KSAEM
μ_λ	0.0236	0.0229 (0.009)	0.0259 (0.013)
μ_ν ($\times$ 10^7)	8.195	8.6327 (4.058)	8.3869 (4.390)
μ_{x_0}	0.4	0.4024 (0.107)	0.5615 (0.200)
ω^2_λ	0.04	0.0391 (0.035)	0.1382 (0.143)
ω^2_ν	0.04	0.0451 (0.050)	0.1688 (0.173)
$\omega^2_{x_0}$	0.04	0.0426 (0.030)	0.0905 (0.113)
σ^2_ε	0.05	0.0391 (0.035)	0.1383 (0.143)

Table 2: Simulation study with KPP model: results obtained from 100 repetitions, with $N = 100$ individuals with the exact SAEM and KSAEM. Results are presented in means and standard deviation in brackets.

computation time and providing reasonable estimates of the parameters.

6 Conclusion

The use of genuine SAEM algorithm on complex models leads to very long computation time. A first idea is to replace the evaluation of the complete model by a simple interpolation on a precomputed grid. This approach requires a long off line step, but SAEM is then very fast. This approach will be interesting if the same model must be run on many different data sets, and if the offline step can take place before the first analysis is necessary.

An other idea is to start from a few precomputed values of the complex model and to complete this basis upon request. In the current approach the off line step is much faster, and the online step is parsimonious. This approach can be useful if there is a few different data sets to analyze.

A drawback of the current algorithm is the necessity to choose the various δ_k. If δ_k decreases too fast, useless evaluations of the complete model will lengthen the computation. On the contrary if δ_k remains too large, the precision of the result of KSAEM will be impaired. To choose optimally δ_k implies to understand the link between the convergence speed of the meta model and the convergence speed of SAEM, a question which is widely open. In our two examples, the choice of δ_k has been done after a few trial and error.

Note that the mathematical proof of the convergence of KSAEM is open

and appears to be a difficult question.

Acknowledgments

This project has been partially funded by the LabEx PERSYVAL-Lab (ANR-11-LABX-0025-01), the PEPS Egalité Integer Project and the Fédération de Recherche en Mathématiques Rhône-Alpes-Auvergne (CNRS FR 3490, Project D04).

References

Auder, B., De Crecy, A., Iooss, B., and Marques, M. (2012). Screening and metamodeling of computer experiments with functional outputs. *Reliability Engineering and System Safety*, 107:122–131.

Barbillon, P., Barthelemy, C., and Samson, A. (2015). Parametric estimation of complex mixed models based on meta-model approach. *Submitted*.

Delyon, B., Lavielle, M., and Moulines, E. (1999). Convergence of a stochastic approximation version of the EM algorithm. *Ann. Statist.*, 27:94–128.

Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977). Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(1):pp. 1–38.

Dupuy, D., Helbert, C., and Franco, J. (2015). DiceDesign and DiceEval: Two R packages for design and analysis of computer experiments. *Journal of Statistical Software*, 65(11):pp. 1–38.

Fang, K., Li, R., and Sudjianto, A. (2005). *Design and Modeling for Computer Experiments (Computer Science & Data Analysis)*. Chapman & Hall/CRC.

Grenier, E., Louvet, V., and Vigneaux, P. (2014). Parameter estimation in non-linear mixed effects models with SAEM algorithm: extension from ODE to PDE. *ESAIM: M2AN*, 48(5):1303–1329.

Haasdonk, B. and Lohmann, B. (2011). Special issue on model order reduction of parameterized problems. *Mathematical and Computer Modelling of Dynamical Systems*, 17(4):295–296.

Jin, R., Chen, W., and Sudjianto, A. (2005). An efficient algorithm for constructing optimal design of computer experiments. *Journal of Statistical Planning and Inference*, 134:268–287.

Kuhn, E. and Lavielle, M. (2005). Maximum likelihood estimation in nonlinear mixed effects models. *Computational Statistics and Data Analysis*, 49(4):1020–1038.

Marrel, A., Iooss, B., Julien, M., Laurent, B., and Volkova, E. (2011). Global sensitivity analysis for models with spatially dependent outputs. *Environmetrics*, 22(3):383–397.

Monolix Team (2012). *The Monolix software, Version 4.1.2. Analysis of mixed effects models. See "The theophylline example" section in the PDF of the Users Guide, eg. www.lixoft.eu/wp-content/resources/docs/UsersGuide.pdf.* LIXOFT and INRIA, http://www.lixoft.com.

Moutoussamy, V., Nanty, S., and Pauwels, B. (2015). Emulators for stochastic simulation codes. *ESAIM: proceedings ans surveys*, 48:116–155.

Patera, A. and Rozza, G. (2006). *Reduced Basis Approximation and A Posteriori Error Estimation for Parametrized Partial Differential Equations.* MIT Monographs (also available online).

Picheny, V. and Ginsbourger, D. (2013). A non-stationary space-time Gaussian process model for partially converged simulations. *Journal of Uncertainty Quantification*, 1:57–78.

Pronzato, L. and Muller, W. (2012). Design of computer experiments: space filling and beyond. *Statistics and Computing*, 22:681–701.

Sacks, J., Schiller, S. B., Mitchell, T. J., and Wynn, H. P. (1989). Design and analysis of computer experiments. *Statistical Science*, 4:409–435.

Santner, T. J., Williams, B., and Notz, W. (2003). *The Design and Analysis of Computer Experiments.* Springer-Verlag.

Schilders, W. H. A., van der Vorst, H. A., and (Eds.), J. R. (2008). *Model Order Reduction: Theory, Research Aspects and Applications*. Springer Berlin Heidelberg.

Wei, G. and Tanner, M. (1990). A Monte Carlo implementation of the EM algorithm and the poor man's data augmentation algorithms. *Journal of the American Statistical Association*, 85:699–704.